FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 31, No 2 (2016), 569–578

APPROXIMATE SPECTRAL LEARNING USING NYSTROM METHOD *

Aleksandar Trokicić

Abstract. Constrained clustering algorithms as an input have a data set and constraints which inform it whether to put two items in the same cluster or not. Spectral clustering algorithms compute cluster assignments from eigenvectors of a matrix that is computed from the data set. In this paper, we study the class of constrained spectral clustering algorithms that incorporate constraints by modifying the graph adjacency matrix. The proposed algorithm combines Nystrom method with the existing spectral learning algorithm to produce a linear (in the number of vertices) time algorithm. We tested our algorithm on real world data sets and we demonstrated that it shows better results on some data sets than the original algorithm. In the end, we propose an algorithm for constrained multi view data clustering.
Keywords: spectral clustering, spectral learning, constrained clustering, Nystrom method

1. Introduction

Clustering is one of the important and well-studied problems in machine learning, data mining and computer vision. The objective of a clustering problem is to form groups of similar or related data. Since large data sets have become easily available, a clustering solution can be used as a rough analysis of a data set or as an input to a classification algorithm. Clustering algorithms can be divided into several categories such as centroid based clustering (each cluster is represented with a point called a centroid and a datum belongs to a cluster with a nearest centroid) like k-means algorithm, distribution based clustering (each cluster represents a distribution) or hierarchical clustering. Clustering can be performed in an original vector space or data can be modified and clustering performed in a different vector space. We focus on spectral clustering algorithms where clustering is performed in a modified vector space with fewer dimensions than the original space. Clustering algorithms belong to a class of unsupervised learning algorithms, only data

Received April 25, 2016; accepted May 08, 2016
2010 *Mathematics Subject Classification.* Primary 68T10; Secondary 62H30
*This research was supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013

values are known at the input. However, in this paper we focus on incorporating pairwise instance level constraints into the clustering algorithm in order to improve performance. In this paper we consider two types of constraints:

- *CL constraints* : two instances cannot be in the same cluster.
- *ML constraints* : two instances must be in the same cluster.

The objective of a constrained clustering algorithm (clustering algorithm that incorporates pairwise instance level constraints) is to partition a data set into groups of similar data while complying with the constraints as much as possible.

We assume that we are given a set of points $\mathbb{R}^N$, similarity measure $\mathcal{S}$ and a set of pairwise constraints. From the set of points and similarity measure we form a similarity graph and cluster its vertices while complying with the constraints.

In this paper we address the constrained spectral clustering problem, the objective of which is to incorporate constraints into the spectral clustering algorithm. Spectral clustering algorithms [8], [11], [10] compute cluster assignments from eigenvectors of a matrix that is computed from the pairwise similarity between data. The prior work on this problem can be divided into three categories. In the first category are algorithms that incorporate constraints by modifying a matrix derived from a graph as in Kamvar et al. [6], Xu et al. [16]. In the second category are algorithms that change the feasible solution space of a spectral clustering algorithm so that it complies with the constraints as in Coleman et al. [4]. In the third category are algorithms that define constrained spectral clustering as a constrained optimization problem as in Wang et al. [14].

Since the complexity of graph matrices construction is $O(n^2)$ and complexity of eigendecomposition is $O(n^3)$ using spectral learning on a large data set is expensive. We suggest the use of Nystrom method [3] [15] [7] to approximate graph matrices and therefore speed up computation. We implemented *approximate spectral learning* algorithm as a combination of spectral learning algorithm from [6] and Nystrom method. For an arbitrary number $l \ll n$ we implemented algorithm such that complexity for graph matrix construction is $O(nl)$ and for eigendecomposition $O(l^3)$ and for final approximation step is $O(nl^2)$. Of course after eigenvector computation both approximate and original algorithm have to perform kmeans algorithms with complexity $O(nt)$ where t is the number of iterations.

We also address the problem of multi-view constrained spectral clustering. It belongs to a class of multi-view learning [2] [5] where input data set consists of multi-view items. In multi-view item we have access to different type of information about an item. For example, one view of an item is image and another view is textual information about that image. Another example of a multi-view item is an image of an object taken with cameras on different locations. Multi-view data set can be gathered using multiple sensors, an information collected by a single sensor represents one view. We implemented an algorithm for constrained multi-view clustering by applying combination of spectral learning algorithm from [6] and Nystrom method on the sum of graph derived matrices for each view.

This paper is organized as follows. In Chapter 2. we explain spectral clustering algorithm. In Chapter 3. we implement approximate spectral learning algorithm as a combination of spectral learning algorithm from [6] and Nystrom method .Application of approximate spectral learning algorithm onmulti-view data set is implemented at the end of chapter 3. In the final two chapters experimental results and conclusion are presented.

2. Spectral Clustering

Spectral clustering forms a partition of graph vertices such that the sum of the edges between the vertices of the same partition is high and the sum of the edges between the vertices of different partitions is low based on eigenvectors of a graph derived matrix. Let $G = (V, E, W)$ be an undirected weighted graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and E is the set of edges. Every edge in the graph is associated with a weight which is a non-negative real number. The adjacency matrix W of the graph is a symmetric matrix where W_{ij} is the edge weight if there is an edge between vertex v_i and v_j, or 0 otherwise. Since the set of edges E can be computed from the adjacency matrix W, we will write $G = (V, W)$. The degree of a vertex v_i is the sum of weights of edges that start in that vertex. The degree matrix D is a diagonal matrix such that D_{ii} is the degree of vertex v_i.

We are given a similarity graph and k as a number of clusters to partition a graph into. The goal of a clustering algorithm is to partition the vertex set into k groups of similar vertices. A natural way to define clustering on a graph is a solution of a RatioCut problem [8]. RatioCut of a cluster (partition) set $(A_1, A_2, \ldots, A_k)$ is the sum $RatioCut(A_{1:k}) = \sum_{i=1}^{i=k} \frac{cut(A_i, V \setminus A_i)}{|A_i|}$, where $cut(A_i, V \setminus A_i)$ measures a total weight of edges from cluster A_i to other clusters. The solution to the RatioCut problem is a set of clusters $(A_1, A_2, \ldots, A_k)$ so that $RatioCut(A_{1:k})$ is minimal. According to [8] relaxing this problem yields:

$$\min_{U \in \mathbb{R}^{n \times k}} \mathrm{Tr}(U^T L U) \quad \text{s. t.} \quad U^T U = I \tag{2.1}$$

where L is a graph Laplacian

$$L = D - W$$

and from a matrix U cluster assignments are derived as a solution of k-means clustering on rows of U. Columns of a matrix U which minimizes the (2.1) are k smallest eigenvectors of a Laplacian matrix L.

In literature there are different normalizations of graph Laplacian used by spectral algorithms such as $D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ [12] and $(W + d_{max} I - D)/d_{max}$ [6] where d_{max} is the maximal degree of a vertex and I is the identity matrix. We will use the normalized additive Laplacian matrix $L_N = (W + d_{max} I - D)/d_{max}$ used by the authors in [6] for spectral clustering.

FIG. 2.1: Spectral clustering. [6]

procedure SC(W, k)
Input: adjacency matrix W ;
Input: number of clusters k ;
Compute the degree matrix D;
Compute the normalized additive Laplacian matrix $L_N = (W + d_{max}I - D)/d_{max}$;
Compute matrix U whose columns are k largest eigenvectors $u_1, \ldots, u_k$ of L_N;
Normalize rows of U;
Cluster rows of U into k clusters using k-means algorithm;
Output: Clusters $C_1, \ldots, C_k$
end procedure

3. Constrained spectral clustering

We are given an input graph $G = (V, W)$ and a set of constraints. We consider two types of constraints:

1. $ML(i, j)$ Vertex v_i and vertex v_j must be in the same cluster.
2. $CL(i, j)$ Vertex v_i and vertex v_j cannot be in the same cluster.

In [6] the problem is solved by changing the adjacency matrix

1. Vertex v_i and vertex v_j must be in the same cluster $\rightarrow$ edge weight is set to 1.
2. Vertex v_i and vertex v_j cannot be in the same cluster $\rightarrow$ edge weight is set to 0.

We will refer to the adjacency matrix with incorporated constraints as modified adjacency matrix W. In the next step spectral clustering (2.1) is applied. The authors of [6] named this algorithm *spectral learning* algorithm. Since the complexity of forming graph matrices is $O(n^2)$ and the complexity of eigendecomposition is $O(n^3)$ using spectral learning on large data sets is expensive. We suggest the use of Nystrom method to speed up computation.

3.1. Nystrom method

We will use Nystrom method as explained in [15]. Nystrom method finds an approximation of a symmetric positive semi-definite matrix $X \in \mathbb{R}^{n \times n}$. An arbitrary number $l \ll n$ is used for approximation. Let C be the matrix of dimensions $l \times n$ whose columns are randomly sampled from X, and let $\mathcal{L}$ be the set of indices of those columns in X. Matrix $Q = X(\mathcal{L}, :)$ is created by sampling rows from

matrix C. Since matrix Q is symmetric we can compute its eigenvalue decomposition $Q = U_Q \Sigma_Q U_Q^T$. Matrix X is approximated with matrix $\tilde{X}$ with the following approximate eigenvalue decomposition

$$\tilde{X} = (\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^{-1})(\frac{n}{l}\Sigma_Q)(\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^{-1})^T$$

The complexity of this method is $\mathcal{O}(nl^2 + l^3)$.

3.2. Spectral learning using Nystrom method

FIG. 3.1: Spectral learning via Nystrom method algorithm.

procedure SL-NYS(W, k, l)
Input: adjacency matrices W ;
Input: number of columns l ;
Input: number of clusters k ;
$\mathcal{L}$ is set of sampled column indices;
$\hat{W} = W(:, \mathcal{L})$;
$\hat{I} = I(:, \mathcal{L})$;
$\hat{D} = D(:, \mathcal{L})$;
$\hat{d}_{max} = max(\hat{D})$;
$C = (\hat{W} + \hat{d}_{max}\hat{I} - \hat{D})/\hat{d}_{max}$;
$Q = C(\mathcal{L}, :) = U_Q \Sigma_Q U_Q^T$;
$\tilde{\Sigma} = \frac{n}{l}\Sigma_Q$;
$\tilde{U} = \sqrt{\frac{l}{n}} C U_Q \Sigma_Q^{-1}$;
Compute U as first k eigenvectors of $\tilde{U}$;
Normalize rows of U
Cluster rows of U into k clusters using k-means algorithm;
Output: Clusters $C_1, \ldots, C_k$
end procedure

We propose to use the algorithm (3.1) for fast approximate spectral learning. Let $\mathcal{L}$ be the set of indices of l randomly sampled columns of modified adjacency matrix W. We assume that adjacency matrix W is modified to include constraints:

- $W_{i,j} = 1$ if $ML(i, j)$ is in the set of constraints,
- $W_{i,j} = 0$ if $CL(i, j)$ is in the set of constraints.

Matrix $\hat{W} = W(:, \mathcal{L})$ is a modified adjacency matrix with sampled columns. In the next step we form matrices $\hat{I} = I(:, \mathcal{L})$ and $\hat{D} = D(:, \mathcal{L})$. Because we do not have access to the entire diagonal matrix, the maximal degree d_{max} of a vertex in graph G, is approximated with $\hat{d}_{max} = max(\hat{D})$.

The following matrix is an approximation of the normalized additive graph Laplacian $C = (\hat{W} + \hat{d}_{max}\hat{I} - \hat{D})/\hat{d}_{max}$. In the next step eigenvalue decomposition of matrix $Q = C(\mathcal{L}, :)$ is computed as $Q = U_Q \Sigma_Q U_Q^T$. It follows that approximated normalized additive graph Laplacian is $\tilde{L} = \tilde{U}\tilde{\Sigma}\tilde{U}^T$ where $\tilde{\Sigma} = \frac{n}{l}\Sigma_Q$ and $\tilde{U} = \sqrt{\frac{l}{n}} C U_Q \Sigma_Q^{-1}$. We use columns of the matrix $\tilde{U}$ as approximated eigenvectors of the normalized graph Laplacian L_N and apply kmeans algorithm on rows of matrix whose columns are largest k eigenvectors of $\tilde{U}$.

3.3. Aproximate spectral learning on multi-view data

Multi-view constrained spectral clustering takes as an input $\{(x_i^1, x_i^2) \in (\mathbb{R}^{N_1} \times \mathbb{R}^{N_2}) \mid i = \overline{1,n}\}$ and a set of must link and cannot link constraints. Vectors x_i^1 and x_i^2 represent two views of the same datum. For example, an image and its textual explanation represent two different views of the same item. Dimensions of views x_i^1 and x_i^2 are not necessarily equal. Let k_1 and k_2 be kernels defined on the first and second view space, respectively. We use direct sum kernel $k((x_i^1, x_i^2), (x_j^1, x_j^2)) = k_1(x_i^1, x_j^1) + k_2(x_i^2, x_j^2)$ to form a similarity graph of the multi-view data set in order to apply the spectral learning algorithm [6].

We propose to implement multi-view approximate spectral clustering algorithm by forming two similarity graphs and to apply Nystrom method on theirs normalized additive Laplacian matrices. In the first step we propose to form two similarity graphs $G_1 = (V_1, W_1)$ and $G_2 = (V_2, W_2)$ for each view induced by k_1 and k_2, respectively, and to modify its adjacency matrices to include constraints. We propose to form an approximate normalized additive graph Laplacians $\hat{C}^1$ and $\hat{C}^2$ from each graph using a set of sampled column indices $\mathcal{L}$. In the next step we form the sum of approximate matrices $\hat{C} = \hat{C}^1 + \hat{C}^2$ and follow the algorithm (3.1) from step (6) to the end.

4. Experimental Results

In this section we evaluate the performance of approximate spectral learning algorithms **SL-NYS** on single-view and multi-view data sets.

4.1. Aproximate spectral learning

We compare the algorithm (3.1) with the original (without approximation) spectral learning algorithm (**SL**) [6] and (**CSP**) from [14]. We implemented the spectral learning algorithm from [6] and our algorithm for approximate spectral learning (3.1) in Matlab and downloaded Matlab code for **CSP** from one of the author's [13].

We use four real world data sets from [1]: Wine, Seeds, Glass and Ionosphere. Constraints are derived from the exact labels:

Table 4.1: The UCI data sets

Data Set	Instances	Attributes
Wine	130	13
Glass	214	9
Seeds	210	7
Ionosphere	351	34

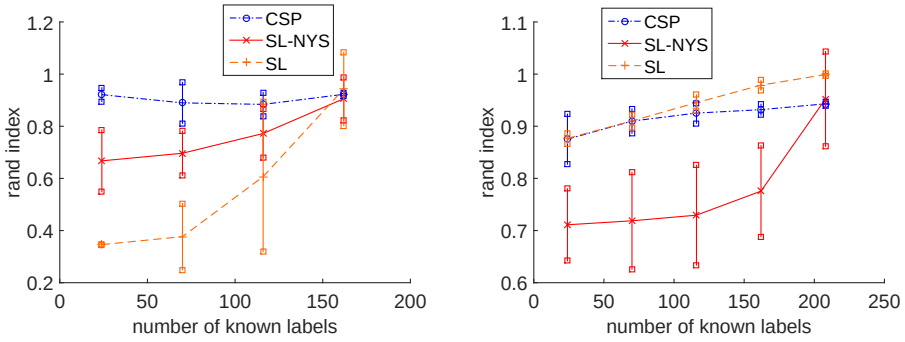
- If item i and item j have the same label we add $ML(i, j)$ to the set of constraints
- If item i and item j have different labels we add $CL(i, j)$ to the set of constraints

For information about data sets see Table (4.1). The adjacency matrix is formed using RBF kernel. The results were evaluated using Rand Index as defined in [9]. In Figure (4.1), x-coordinate represents the number of known instances and Y-axis represents the rand index. We tested on a number of known labels between 24 and the size of the data set. At each stop we performed 31 tests where we randomly selected the known items and reported mean and variance of the results.

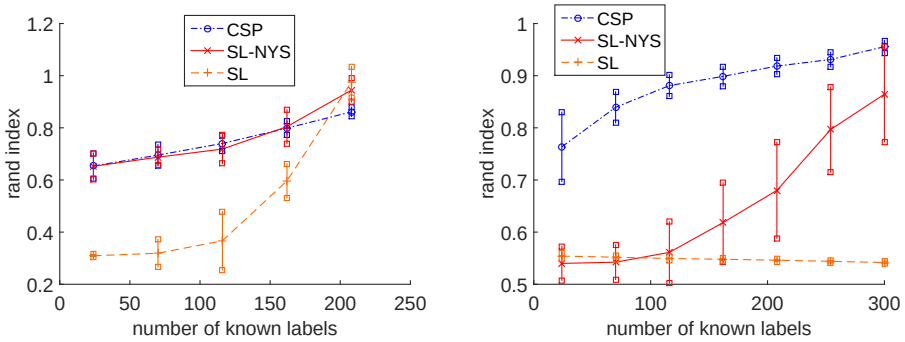
Let us look at the results form Figure (4.1). On three data sets Glass, Ionosphere and Wine, the approximate spectral learning algorithm produced mostly better results than the original spectral learning algorithm and on Seeds data set the original spectral learning algorithm mostly performed better. However, the approximate spectral learning algorithm produced similar results to **CSP** only on Glass data set and **CSP** produced better results on the rest of data sets. The original spectral learning algorithm performed better than **CSP** on Seeds data set, while **CSP** performed better on the rest of data sets.

4.2. Approximate multi-view spectral learning

Data sets Wine and Glass are used for testing the approximate multi-view spectral learning algorithm. We will test algorithms on 2 view data sets. The elements (feature vectors) of both Glass and Wine data sets are single view items. Therefore, we need to convert them into 2-view items. We split each feature vector into two feature vectors in a way that the first half of the features represents the first view and the second half of the features represents the second view. Direct sum kernel is used as an input to the **CSP** algorithm and the spectral learning algorithm [6]. Approximate spectral learning produced better results than the original spectral learning algorithm on Glass data set. On Wine data set for a small number of known labels the approximate algorithm produced better results than the original algorithm and the situation was reverse when the number of known labels was large. **CSP** produced mostly better results than other algorithms.

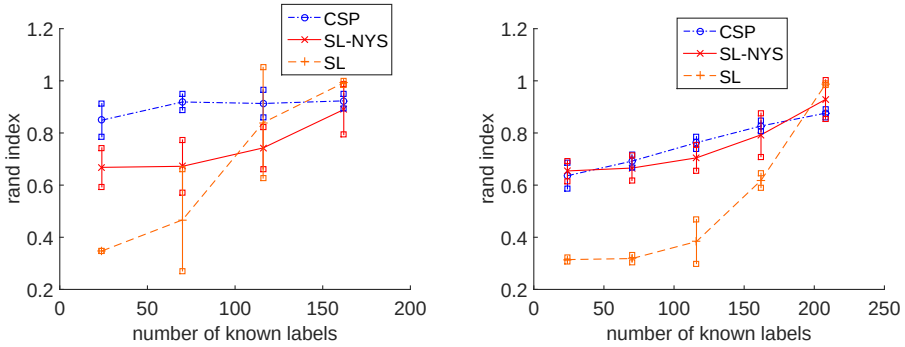


(a) Wine (b) Seeds



(c) Glass (d) Ionosphere

FIG. 4.1: Performance comparison of different spectral clustering algorithms on real world data sets with constraints.



(a) Wine (b) Glass

FIG. 4.2: Performance comparison of different spectral clustering algorithms on real world multi-view data sets with constraints.

5. Conslusion

In this paper we implemented the approximate spectral learning algorithm (3.1). It is an approximation of the spectral learning algorithm [6] using Nystrom method. We compared the algorithm (3.1) with the original spectral learning algorithm and on some data sets the approximated algorithm performed better. Even though the algorithm (3.1) produced better results than the original one it did not perform better than **CSP** [14]. We also showed how to apply our algorithm to a multi-view data set. Empirically, it showed better results than the original spectral learning algorithm [6] on a small number of known labels. However, **CSP** [14] mostly produced the best results. The advantage of the approximate spectral learning algorithm over the other two algorithms is its complexity. The complexity of the algorithm (3.1) is linear in the number of vertices while on the other hand the spectral learning algorithm [6] and **CSP** [14] have quadratic complexity.

REFERENCES

1. K. Bache and M. Lichman: *UCI Machine Learning Repository.* [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine, CA, 2013.
2. A. Blum and T. Mitchell: *Combining labeled and unlabeled data with co-training.* In: Proceedings of the Conference on Computational Learning Theory (Morgan Kaufmann, eds.), 1998, pp. 92 - 100.
3. A. Choromanska, T. Jebara, H. Kim, M. Mohan and C. Monteleoni: *Fast Spectral Clustering via the Nystrm Method.* Algorithmic Learning Theory, Lecture Notes in Computer Science **8139** (2013), pp 367–381.
4. T. Coleman, J. Sanderson and A. Wirth: *Spectral clustering with inconsistent advice.* In: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 152 - 159.
5. E. Eaton, M. desJardins and S. Jacob: *Multi-view constrained clustering with an incomplete mapping between views.* Knowledge and Information Systems, **38** (2012), pp 231 – 257.
6. S. D. Kamvar, D. Klein and C. D. Manning: *Spectral learning.* In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp 561–566.
7. S. Kumar, M. Mohri and A. Talwalkar: *Sampling techniques for the Nystrom method.* Journal of Machine Learning Research, **5** (2013), pp 304–311.
8. U. von Luxburg: *A tutorial on spectral clustering.* Statistics and Computing, **17(4)** (2013), pp 395–416.
9. C. D. Manning, P. Raghavan and H. Schutze: *Introduction to Information Retrieval.* Cambridge University Press, 2008.
10. M. Meila, and J. Shi: *Learning Segmentation by Random Walks.* In: Advances in Neural Information Processing Systems, **13** (2001), pp 873–879.

11. A. Ng, M. Jordan and Y. Weis: *On Spectral Clustering: Analysis and an algorithm*. In: Advances in Neural Information Processing Systems, **14** (2001), pp 849–856.
12. J. Shi, and J. Malik: *Normalized cuts and image segmentation*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, pp 888 - 905.
13. X. Wang: [https://sites.google.com/site/gnaixgnaw/home]
14. X. Wang, B. Qian and I. Davidson: *On Constrained Spectral Clustering and Its Applications*. Data Mining and Knowledge Discovery, **28(1)** (2014), pp 1 – 30.
15. C. Williams and M. Seeger: *Using the Nystrom method to speed up kernel machines*. In: NIPS, (2000), pp 682 – 688.
16. Q. Xu, M. desJardins and K. Wagstaff: *Constrained spectral clustering under a local proximity structure assumption*. In: Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference, 2005, pp 866 - 867.

Aleksandar Trokić
Faculty of Sciences and Mathematics
Department of Computer Science
P.O. Box: 224
18000 Niš, Serbia
aleksandar.trokicic@pmf.edu.rs